Claims

1. A system comprising means for sorting incoming data packets in real time; means for assigning an exit order to said packets in real time; and queue means for receiving said sorted packets in said exit order.

2. A system as claimed in claim 1 wherein the sorting means is responsive to information contained within a packet whereby to determine an exit order number for that packet.

3. A system as claimed in claim 1, wherein the sorting means is responsive to information contained in a table whereby to determine an exit order number for that packet.

4. A system as claimed in claim 1, wherein the sorting means is responsive to information associated with a data packet stream in which said packet is located whereby to determine an exit order number for that packet.

5. A system as claimed in claim 1, wherein said sorting means is adapted to insert sorted packets in said queue means in exit order.

6. A system as claimed in claim 1 or 5, wherein said queue means is a single queue.

7. A system as claimed in claim 6, wherein said single queue provides a plurality of virtual queues.

8. A system as claimed in claim 1, further comprising a queue manager adapted to insert packets into said queue means in exit order.

9. A system as claimed in claim 1, further comprising means to drop certain packets before being output from said queue means.

10. A system as claimed in claim 1, further comprising means to drop certain packets before being queued in said queue means.

BEST AVAILABLE COPY

11. A system as claimed in any of the preceding claims, wherein:

said sorting means and said queue means process only packet records containing information about said packets, and

data portions of said packets are stored for output in accordance with an exit order determined for the corresponding packet record.

12. A system as claimed in any of the preceding claims, wherein said sorting means comprises a parallel processor.

13. A system as claimed in claim 12, wherein said parallel processor is an array processor.

14. A system as claimed in claim 12, wherein said array processor is a SIMD processor.

15. A system as claimed in claim 12, 13 or 14, further comprising means to provide access for said parallel processors to shared state.

16. A system as claimed in claim 15, further comprising a state engine to control said access to said shared state.

17. A system as claimed in any of claims 1 to 16, further comprising tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor or to each processor element of a parallel processor.

18. A system as claimed in claim 17, wherein said tables are the same on each processor or on each processor element of a parallel processor.

19. A system as claimed in claim 17, wherein said tables are different on different processors or on different processor elements of a parallel processor.

20. A system as claimed in claim 17, wherein said processors or processor elements share information from their respective tables, such that:

(a) the information held in the table for one processor is directly accessible by a different processor or the information held in the table in one processor element is accessible by other processing element(s) of the processor; and

BEST AVAILABLE COPY AMENDED SHEET

(b) processors have access to tables in other processors or processor elements have access to other processor elements in the processor, whereby processors or processor elements can perform table lookups on behalf of other processor(s) or processor elements of the processor.

21. A system as claimed in any of the preceding claims, wherein said sorting means implement algorithms for packet scheduling in accordance with predetermined criteria, such as WFQ, DFR, congestion avoidance (eg WRED) or other prioritisation and sorting.

22. A method for sorting incoming data packets in real time, comprising sorting the packets into an exit order; and

queueing said sorted packets for output in said exit order.

23. A method as claimed in claim 22, wherein the sorting is responsive to information contained within a packet whereby to assign an exit order number for that packet.

24. A method as claimed in claim 22, wherein the sorting is responsive to information contained in a table whereby to determine an exit order number for that packet.

25. A method as claimed in claim 22, wherein the sorting is responsive to information associated with a data packet stream in which said packet is located whereby to determine an exit order number for that packet.

26. A method as claimed in claim 22, wherein said packets are inserted into a queue means in exit order determined by the means performing the sorting.

27. A method as claimed in claim 22, comprising inserting sorted packets into a queue means in exit order under control of a queue manager.

28. A method as claimed in claim 26 or 27, wherein said queueing is performed using a single output queue.

29. A method as claimed in claim 28, further comprising providing a plurality of virtual queues by means of said single output queue.

BEST AVAILABLE COPY

30. A method as claimed in claim 22, further comprising dropping certain packets before being output from said queue means.

31. A method as claimed in claim 22, further comprising dropping certain packets before being queued in said queue means.

32. A method as claimed in any of claims 22 to 31, wherein:

said sorting and said queuing operations are performed only on packet records containing information about said packets, said method further comprising:

storing data portions of said packets in said memory for output in accordance with an exit order number determined for the corresponding packet record.

33. A method as claimed in any of claims 22 to 31, wherein said sorting is performed by a parallel processor.

34. A method as claimed in claim 33, wherein said parallel processor is an array processor.

35. A method as claimed in claim 33, wherein said array processor is a SIMD processor.

36. A method as claimed in claim 33, 34 or 35, further comprising providing access for said processors to shared state under control of a state engine.

37. A method as claimed in claim 36, further comprising providing tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor or to each processor element of a parallel processor.

38. A method as claimed in claim 37, wherein said tables are the same on each processor or on each processor element of a parallel processor.

39. A method as claimed in claim 37, wherein said tables are different on different processors or on each processor element of a parallel processor.

40. A method as claimed in claim 37, wherein said processors or processor elements share information from their respective tables, such that:

BEST AVAILABLE COPY

(a) the information held in the table for one processor is made directly accessible by a different processor or the information held in the table of one processor element is made directly accessible to other processor element(s) of the processor; and

(b) access is provided for said processor or processor elements to tables in other processors or processor elements, whereby processors or processor elements can perform table lookups on behalf of another processor or processor element.

41. A system as claimed in any of claims 1 to 21, wherein said sorting means implement algorithms for packet scheduling in accordance with predetermined criteria, such as WFQ, DFR, congestion avoidance (eg. WRED) or other prioritisation and sorting.

42. A computer system, comprising a data handling system as claimed in any of claims 1 to 21.

43. A network processing system, comprising a data handling system as claimed in any of claims 1 to 21.

44. A computer system adapted to perform the method as claimed in any of claims 22 to 40.

45. A network processing system adapted to perform the method as claimed in any of claims 22 to 40.

46. A computer system as claimed in claim 42 implemented as one or more silicon integrated circuits.

47. A data carrier containing program means adapted to perform the method as claimed in any of claims 22 to 40.

BEST AVAILABLE COPY